

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. An algorithm to improve efficiency of editing source code, comprising
 - (1) recognizing that source code has been edited;
 - (2) identifying a program construct having the edited source code;
 - (3) constructing a construct list of at least one other construct having similar and/or related code;
 - (4) determining the similarity between the at least one other construct and the program construct having the edited source code;
 - (5) if equal to or beyond a threshold of similarity, then notifying the owners of the at least one other construct determined to be similar.
2. The efficiency algorithm of claim 1, wherein the step of identifying the program construct further comprises parsing the tokens of the edited source code.
3. The efficiency algorithm of claim 1, wherein the step of constructing a construct list further comprises determining that the at least one other construct is of a reasonable size for placement in the construct list.
4. The efficiency algorithm of claim 3, further comprising the step of: parsing a sequence of tokens from each of a plurality of constructs of a reasonable size.
5. The efficiency algorithm of claim 4, wherein the step of determining the similarity further comprises comparing the parsed tokens of the edited source code with the parsed tokens of each of a plurality of constructs in the construct list.
6. The efficiency algorithm of claim 5, wherein the step of comparing the parsed tokens further comprises weighting the compared tokens so that a degree of similarity can be established.

7. The efficiency algorithm of claim 6, further comprising the step of summing the weights of the compared tokens to determine if the sum is equal to or beyond the threshold of similarity.
8. The efficiency algorithm of claim 1, further comprising storing the construct list.
9. The efficiency algorithm of claim 1, wherein the efficiency algorithm is a machine-implemented process in an integrated development environment.
10. An efficiency algorithm to improve efficiency of editing source code, comprising
 - (1) recognizing that source code has been edited;
 - (2) identifying a program construct having the edited source code and parsing the tokens of the edited source code;
 - (3) constructing a construct list of at least one other construct of reasonable size having similar and/or related code by parsing a sequence of tokens from each of a plurality of constructs of a reasonable size;
 - (4) determining the similarity between the at least one other construct and the program construct having the edited source code by comparing the parsed tokens of the edited source code with the parsed tokens of each of a plurality of constructs in the construct list, and weighting the compared tokens;
 - (5) summing the weights of the compared tokens to determine if the sum is equal to or beyond the threshold of similarity, and if so, then determining if an owner of the at least one other construct determined to be similar is to be notified; and
 - (6) storing the construct list.
11. A method of tracing determining if two or more constructs in a repository of source code in an integrated development environment are related and/or derived, said method comprising the steps of:
 - (1) identifying a first construct;
 - (2) parsing the first N tokens of the first construct;

- (3) identifying a plurality of other constructs in the repository;
 - (4) disregarding those of the plurality of constructs in the repository that are too small;
 - (5) identifying those of the plurality of constructs in the repository that are too large;
 - (6) finding subconstructs in the plurality of constructs that are too large;
 - (7) identifying those constructs in the repository and those subconstructs that are of a reasonable size;
 - (8) comparing N tokens of the reasonably sized constructs with N tokens of the first construct;
 - (9) determining a weight for each token based on name, type, and/or representation;
 - (10) summing the weights of each of the N compared tokens;
 - (11) determining that the sum of the weights of the compared token meets or exceeds a threshold of similarity; and
 - (12) determining that the reasonably sized construct having the sum of the weights that meets or exceeds the threshold of similarity is related to the first construct.
12. The method of claim 11, wherein the step of identifying the first construct further comprises identifying that source code within the first construct has been edited.
 13. The method of claim 11, further comprising storing a pointer to the reasonably sized construct having the sum of the weights that meets or exceeds the threshold of similarity in a construct list of related construct.
 14. The method of claim 13, further comprising allocating ownership of a plurality of owners, each of the plurality of owners associated with each of the constructs in the construct list.
 15. The method of claim 14, further comprising: offering notification to the plurality of owners that one of the constructs in the construct list has been changed.

16. An integrated development environment, comprising:
- (1) a repository of source code comprising programs in the integrated development environment;
 - (2) a constructor to determine that within an edited program, a construct has been edited;
 - (3) a construct list within the repository, the construct list having any constructs in the repository of at least N tokens and smaller than P tokens that are similar and/or related to the edited construct;
 - (4) a parser to parse the edited construct and the similar and/or related constructs;
 - (5) a matchmaker that weighs the similarities between the edited construct and the similar and/or related constructs and determines a degree of matching between the similar and/or related constructs in the construct list based on weight assigned to each of the tokens; and
 - (6) an announcer to announce to any of a plurality of programmers accessing the integrated development environment that the edited construct has been edited and that certain other constructs in the construct list have a degree of matching equal to or exceeding a threshold of the edited construct.
17. The integrated development environment of claim 16, further comprising: the ability to list within the construct list all source derived from the edited construct and/or any of the other constructs in the construct lists that have a degree of matching equal to or exceeding a threshold of the edited construct.
18. An integrated development environment, comprising:
- (1) means to create a construct list, the construct list having pointers to constructs within source code being developed within the integrated development environment and related to and/or derived from other constructs in the construct list;
 - (2) means to edit a construct in the construct list;
 - (3) means to determine the effect of the change of the edited construct on the other constructs in the construct list;

- (4) means to notify owners of other constructs in the construct list that the change may have an effect on the other constructs in the construct list.
- 19. The integrated development environment of claim 18, further comprising:
 - (1) means to add a new construct to the construct list if the new construct is derived from one of the constructs in the construct list.
- 20. An article of manufacture, comprising a data storage medium tangibly embodying a program of machine readable instructions executable by an electronic processing apparatus to perform method steps for operating an electronic processing apparatus, said method steps comprising the steps of:
 - (1) determining that source code has been edited in an environment of computer program development;
 - (2) determining if the edited source code is within a construct of size larger than M tokens and smaller than N tokens;
 - (3) parsing the construct having the edited source code;
 - (4) finding and parsing other constructs in the environment having a size larger than M tokens and smaller than N tokens;
 - (5) creating a construct list of other constructs in the environment having a size larger than M tokens and smaller than N tokens;
 - (6) comparing the tokens between the construct having the edited source code and the tokens of the other constructs in the construct list; and
 - (7) determining that the construct having the edited source code is similar to another of the constructs in the construct list.
- 21. The article of manufacture of claim 20, further comprising weighting the compared tokens based on value, type, and/or representation.

22. The article of manufacture of claim 20, said method steps further comprising the step of notifying the owner of the another construct in the construct list that is similar to the edited construct of the similarity and of the changed source code in the edited construct.
23. The article of manufacture of claim 20, further comprising at least one construct list of related and/or derived constructs within the integrated development environment.